Precision Requirements for Quotient Digit Selection in High-Radix Division

Behrooz Parhami
Dept. Electrical and Computer Enginnering
University of California
Santa Barbara, CA 93106 USA
parhami@ece.ucsb.edu

Abstract

Digit-recurrence binary dividers are sped up via two complementary methods: keeping the partial remainder in carry-save form and selecting quotient digits in a radix higher than 2, usually in redundant form. The redundancy provides some tolerance to imprecision, so that the quotient digits can be selected based on examining truncated versions of the partial remainder and divisor. No closed form formula for the required precision in the partial remainder and divisor, as a function of the quotient digit set and the partial remainder range, is known. We establish upper bounds on the required precision for the partial remainder and divisor. The bounds are tight in the sense that each is only one bit over a well-known lower bound.

Guide to Notation

$[\bullet, \bullet]$	Interval of (integer) values, inclusive at both ends
lb/ub	Lower/upper bound; used as lb(•) or ub(•)
a	Maximum magnitude of a radix-r quotient digit
d, d_T	Divisor, and its truncated form
h	Partial remainder has magnitude at most $h\lvert d\rvert$ $(h < 1)$
p, p_T	Shifted partial remainder (rs) and its truncated form
q	Quotient; fractional value with digits denoted as q_{-j}
r	Number representation radix
$s, s^{(j)}$	Partial remainder in general, and after the jth step
x	Candidate radix-r quotient digit
z	Dividend; fractional value with digits denoted as z_{-j}
δ	Precision required of the divisor
ε	Precision required of the partial remainder

1. Introduction

Division is the most complex, and hence slowest, of the four basic arithmetic operations. Thus, techniques for speeding up division have been extensively studied by researchers in computer arithmetic [Parh00], [Parh02]. Simple dividers are usually based on digit-recurrence algorithms employing repeated additions, while high-performance implementations tend to use multiplier-based convergence schemes [Flyn01]. In radix-r digit-recurrence division [Erce94], digits of the quotient are derived one at a time, from the most to the least significant. For a k-digit quotient, there are at least k cycles in the division process; one cycle for producing each radix-r quotient digit, plus additional cycles for initialization, possible final correction, and rounding. There are but two ways to speed up such an algorithm: reducing the number of cycles and making each cycle shorter. The first implies higher radix division, while the second leads to several techniques, including keeping the partial remainder in carry-save form to eliminate the long delay of carry-propagate addition in each cycle.

To make the use of carry-save partial remainder possible, quotient digits may be chosen from a redundant set, such as [–2, 2] in radix 4. The redundancy provides some tolerance to imprecision, so that the quotient digits can be selected based on examining truncated versions of the divisor and the partial remainder [Robe58], [Atki68]. A carry-save partial remainder, truncated to a few bits, can be quickly transformed into an estimate for the partial remainder or its bits used directly as inputs to a PLA or ROM table that spews out quotient digit values. Among other issues, the designer of a fast digit-recurrence divider determines the precision with which the divisor and the partial remainder must be examined to deduce the value of the next quotient digit. This precision, stated in number of bits before and after the radix point, dictates the complexity of the PLA or the size of the ROM table yielding the quotient digit value. The PLA/ROM size influences not only the cost (VLSI area) but also the circuit speed, given that the quotient digit selection block is usually on the critical path.

Many practical aspects of the quotient digit selection process and associated hardware implementations have been discussed in the literature [Zura87], [Erce94], [Ober97], [Flyn01]. However, no closed form formula for the required precision in the divisor and the partial remainder, as a function of the quotient digit set and the range of the partial remainder, is known. So the design process is often described as trial and error. In this paper, we establish tight upper bounds on the required precision for the divisor and the partial remainder. The bounds are tight in the sense that each is only one bit over a well-known simple lower bound, leading to the requirement for just four trial points during the design process: the one represented by the lower bounds, one (or two) more bit(s) of precision for d, an extra bit of precision for p. If none of these four points leads to a viable design, then one more bit of precision for both d and p is guaranteed to work.

0-7803-7147-X/01/$10.00©2001 IEEE

2. Quotient Digit Selection

Radix-r digit-recurrence division for the (adjusted) dividend $z = 0.z_{-1}z_{-2} \ldots z_{-k} = s^{(0)}$ and the normalized divisor $d = 0.1d_{-2}d_{-3} \ldots d_{-k}$ is based on the equation

$$s^{(j)} = rs^{(j-1)} - q_{-j}d$$

where $s^{(j)}$ is the jth partial remainder and $q = 0.q_{-1}q_{-2} \ldots q_{-k}$ is the quotient. We assume $s^{(0)} = |z| \leq h|d|$, perhaps ensured through preshifting of the dividend, yielding the adjusted dividend. The parameter h will be defined shortly. Additionally, we assume that the radix is a power of 2, so that everything is done in binary, with a radix-r digit corresponding to $\log_2 r$ bits.

To allow the choice of q_{-j} based on truncated versions of d and $p^{(j)} = rs^{(j)}$, we use a redundant quotient digit set $[-a, a]$, with $a \geq r/2$ (usually, we have $a \leq r - 1$, but overredundant digit sets satisfying $a \geq r$ are also of some interest and have been used in practice).

Assuming $d > 0$, to maintain the invariant $|s| \leq hd$ throughout the division process, we must be able to restore a shifted partial remainder, having the worst-case magnitude rhd, to hd or less by subtracting ad from it, thus leading to the requirement:

$$rhd - ad \leq hd \quad \text{or} \quad h \leq a/(r-1)$$

Choosing $h = a/(r - 1)$, so as to impose the least possible restriction on the range of s, we can easily determine the boundaries of the region where x is a valid quotient digit:

$$-hd \leq p - xd \leq hd$$

This leads to the following range for p, across which x is a valid quotient digit value:

$$(x-h)d \leq p \leq (x+h)d$$

The ranges associated with the validity of x and $x - 1$ as quotient digit values overlap. The values of p for which both x and $x - 1$ are valid quotient digit choices satisfy:

$$(x-h)d \leq p \leq (x-1+h)d$$

Figure 1 is a graphical representation of this overlap region which is bounded by the two straight lines $p = (x - h)d$ and $p = (x - 1 + h)d$. This representation is known as a p-d plot. Within the overlap region shown in the partial p-d plot of Fig. 1, the quotient digit can be chosen to be $x - 1$ or x. Thus, the boundary for choosing between $x - 1$ and x as quotient digit value can be drawn anywhere within the overlap region.

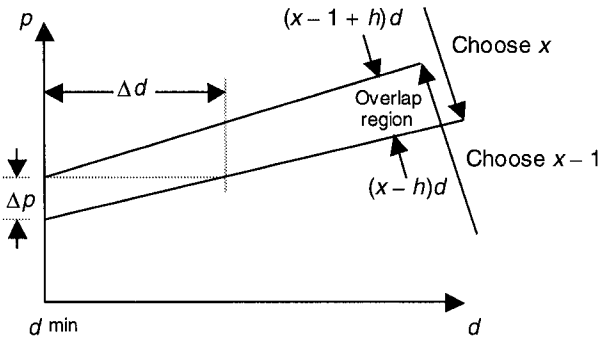

Fig. 1. Part of a p-d plot, showing an overlap region along with Δd and Δp.

3. Known Lower Bounds for Precision

If we truncate d to δ bits after the radix point, the uncertainty in its value, based on the truncated form d_T, will be less than $w = 2^{-\delta}$. In other words, $d_T \leq d < d_T + 2^{-\delta}$. Similarly if we truncate p to ε bits after the radix point to get p_T, we have $p_T \leq p < p_T + 2^{-\varepsilon}$. These two truncations, together define a grid or tiling on the p-d plot, with vertical (horizontal) grid lines being $w = 2^{-\delta}$ ($y = 2^{-\varepsilon}$) apart.

Choice of quotient digit based on d_T and p_T is possible only if none of the rectangular $w \times y$ tiles on the p-d plot, each one representing uncertainties in the values of d and p, intersects both boundaries of an overlap region. Put another way, between each pair of adjacent vertical (horizontal) grid lines, there must exist a horizontal (vertical) grid line that is totally contained in the overlap region. Defining Δd as the minimum horizontal separation between boundaries of the overlap region (Fig. 1), clearly we must have $w \leq \Delta d$. Based on the equations for the two lines bounding the overlap region in Fig. 1, we find

$$\Delta d = d^{\min}(2h - 1) / (x - h)$$

which assumes its smallest possible value for $x = a$. Hence, we arrive at the following well-known lower bound (lb) for the precision required of d in quotient digit selection:

$$\delta \geq \text{lb}(\delta) = \lceil -\log_2 \Delta d \rceil = \lceil -\log_2 [d^{\min}(2h-1)/(a-h)] \rceil$$

A similar argument yields:

$$\varepsilon \geq \text{lb}(\varepsilon) = \lceil -\log_2 \Delta p \rceil = \lceil -\log_2 [d^{\min}(2h-1)] \rceil$$

Of course, in addition to the fractional bits of d and p, as dictated by the bounds above, integer bits of each operand must also be inspected. Because even in unsigned division, the partial remainder can go negative, the sign bit of p is important as well. In the case of signed division, the sign bit of d is also involved in the decision.

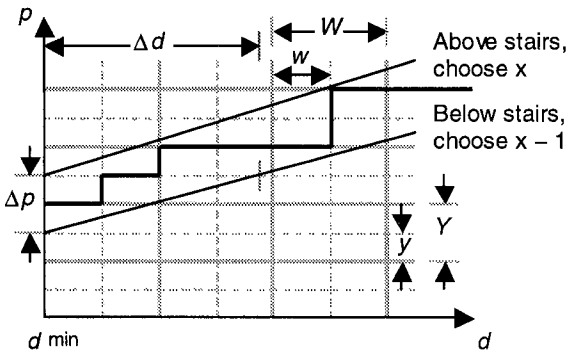

Fig. 2. Tiling the *p-d* plot and embedding stairs in an overlap region.

Returning to the discussion of the number of fractional bits needed, we endeavor to prove that setting $\delta = \text{lb}(\delta) + 1$ and $\varepsilon = \text{lb}(\varepsilon) + 1$ is always sufficient for quotient digit selection, thus establishing the simultaneous upper bounds $\text{ub}(\delta) = \text{lb}(\delta) + 1$ and $\text{ub}(\varepsilon) = \text{lb}(\varepsilon) + 1$. Figure 2 represents an example. Here, based on the precision lower bounds provided by Δd and Δp, the coarse grid of rectangular $W \times Y$ tiles (solid, heavy gray lines) is built. The resulting precisions are clearly inadequate, as evident from the absence of a solid horizontal grid line spanning the overlap region between the vertical grid lines at $d^{\min}$ and $d^{\min} + W$. Our results will show that halving the grid line spacing in each dimension (adding the broken grid lines in Fig. 2) provides adequate precision in the worst case.

4. Upper Bounds for Precision

With the background presented in the previous sections, we are now ready to state and prove our main result.

Theorem 1: To select a quotient digit in $[-a, a]$ for radix-r division, with r a power of 2, it is sufficient to inspect δ bits of d and ε bits of p after their radix points, where:

$$\delta = \lceil -\log_2[d^{\min}(2h - 1)/(a - h)] \rceil + 1$$

$$\varepsilon = \lceil -\log_2[d^{\min}(2h - 1)] \rceil + 1$$

Proof: We need to show that if vertical grid lines are spaced $w = 2^{-\delta}$ apart, there will be at least one horizontal grid line that is entirely contained in the overlap region between two successive vertical grid lines. The proof is based on Fig. 3 and consists of showing that any rectangle that touches two consecutive vertical grid lines and the edges of the overlap region has a height of at least $\Delta p/2$, which is no less than $2^{-\varepsilon}$, the spacing between horizontal grid lines. We only need to prove this at the leftmost edge of the uppermost overlap region (defining the boundary between choosing $a - 1$ or a as the quotient digit value). Corresponding rectangles in other parts of the same overlap region or in other overlap regions are taller, as exemplified by the rectangle near the right end of the overlap region in Fig. 3. So, we want to show that $u \geq \Delta p/2$. But this is a simple consequence of $w \leq \Delta d/2$, which leads to $v \leq \Delta p/2$.

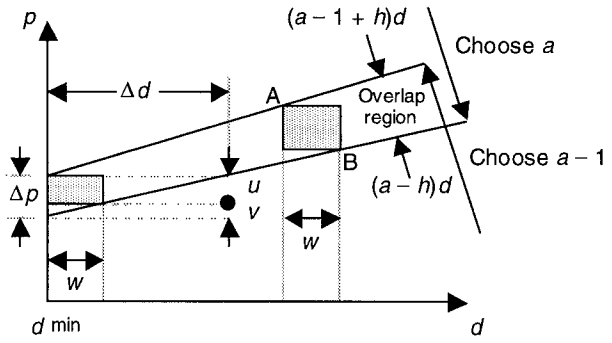

Fig. 3. Part of a *p-d* plot and notation used in the proof of Theorem 1.

5. Determining the Required Precision

Based on the results of Theorem 1, the following procedure should be used to determine the precision required of d and p for selecting the quotient digit. First we note that only the four cases shown in Table I need to be investigated. If none of these four cases proves viable, then the use of $\text{lb}(\delta) + 1$ bits of precision for d and $\text{lb}(\varepsilon) + 1$ bits of precision for p (per Theorem 1) is warranted.

Table I. Cases to be tried before using the upper bounds of Theorem 1.

Case	δ	ε	Notes
1	$\text{lb}(\delta)$	$\text{lb}(\varepsilon)$	Begin with lower bounds
2	$\text{lb}(\delta)+1$	$\text{lb}(\varepsilon)$	Increase the precision of d first, because it is easier than p, which is in carry-save form
3	$\text{lb}(\delta)$	$\text{lb}(\varepsilon)+1$	Next, increase the precision of p
4	$\text{lb}(\delta)+2$	$\text{lb}(\varepsilon)$	Inspecting 2 bits of d is simpler than 1 bit each of d and p (carry-save); so try this case too

Verification in each case is easy to mechanize. The process consists of checking that for all of the overlap regions, of which there are a in the first quadrant, and all consecutive pairs of vertical grid lines, of which there are $2^{\delta}(d^{\max} - d^{\min})$, a horizontal grid line exists that is totally contained in the overlap region. Referring to Fig. 3, this means that for the points A and B of the rectangle associated with the overlap

region for $x - 1$ and x, and the pair of vertical grid lines at $d^{min} + 2^{-\delta}m$ and $d^{min} + 2^{-\delta}(m + 1)$, the coordinates

$$p(\mathrm{A}) = (x - 1 + h)(d^{min} + 2^{-\delta}m)$$

$$p(\mathrm{B}) = (x - h)(d^{min} + 2^{-\delta}(m + 1))$$

must be such that there exists an integer multiple of $2^{-\varepsilon}$ between them; i.e., we must have:

$$\lfloor 2^{\varepsilon}p(\mathrm{A}) \rfloor \geq 2^{\varepsilon}p(\mathrm{B})$$

So, the algorithm, presented in Fig. 4, has two nested loops, corresponding to varying x and m, with the inner loop termination condition being $2^{\varepsilon}p(\mathrm{B}) - 2^{\varepsilon}p(\mathrm{A}) \geq 1$. This latter condition is sufficient for the existence of the required horizontal grid line, but is not necessary; for example, there exists an integer between 2.75 and 3.25, even though their difference is less than 1. Note that the condition is verified only in the first quadrant of the p-d plot, for even though the choice of quotient digit is not symmetric about the two axes, the placement of the uncertainty regions, which affects the required precisions, is symmetric.

```
function adequate(r,a,d^min,d^max,δ,ε)
h := a/(r − 1)
x := 1                                  /* Index of overlap region */
while x ≤ a do                          /* Check the x−1 / x overlap*/
    m := 0                              /* Index of vertical grid line */
    n := 2^δ( d^max − d^min)
    A := 2^ε (x − 1 + h)d^min           /* Initial value for 2^ε p(A) */
    B := 2^ε (x − h)(d^min + 2^−δ)      /* Initial value for 2^ε p(B) */
    S := 2^ε−δ (x − 1 + h)              /* Step size for 2^ε p(A) */
    T := 2^ε−δ (x − h)                  /* Step size for 2^ε p(B) */
    while m < n do                      /* Btwn mth & (m+1)th lines */
        if A − B ≥ 1
        then m := n                     /* No need to check further */
        else if ⌊A⌋ ≥ B
            then m := m + 1             /* Check next grid line */
            else return false           /* Precision is inadequate */
            endif
        endif
        A = A + S
        B = B + T
    endwhile
    x := x + 1
endwhile
return true                             /* Precision is adequate */
endfunction adequate
```

Fig. 4. Verifying if precisions of δ bits for d_T and ε bits for p_T are adequate.

Note that for the sake of clarity, the pseudocode in Fig. 4 has not been fully optimized. In an actual coding of the algorithm, the values of A, B, S, and T could be initialized for $x = 1$ outside the x while loop and then appropriately updated with advancing x. Updating of these values inside the while loop would then require only addition of increment values, in much the same way that updating of A and B in the inner m loop is now handled.

6. Conclusion

In the literature on computer arithmetic, it is often mentioned that determining the precisions required of truncated forms of d and p to correctly choose the radix-r quotient digits from the digit set $[-a, a]$ involves several iterations over the design space. In this paper, we have proven that the number of cases to be tried is limited to at most four, as listed in Table 1, given choices for r (power-of-2 radix) and a (defining a symmetric, redundant digit set). Considering that the trials are easily mechanized and that the choice of the radix r is quite limited by cost/performance requirements, the design process is not as cumbersome as one might have thought.

A question for continued research is whether the number of cases to be tried can be further reduced. For example, one migth investigate if any of the cases listed in Table I can be ruled out based on simple analytical tests (as opposed to exhaustive testing). However, this is only of theoretical interest, as testing based on the algorithm given in Fig. 4 is simple enough for practical purposes.

References

[Atki68] Atkins, D.E., "Higher Radix Division Using Estimates of the Divisor and Partial Remainders," *IEEE Trans. Computers*, Vol. 17, No. 10, pp. 925-934, 1968.

[Erce94] Ercegovac, M.D. and T. Lang, *Division and Square Root: Digit-Recurrence Algorithms and Implementations*, Kluwer Academic, 1994.

[Flyn01] Flynn, M.J. and S.F. Oberman, *Advanced Computer Arithmetic Design*, Wiley, 2001.

[Ober97] Oberman, S.F. and M.J. Flynn, "Division Algorithms and Implementations," *IEEE Trans. Computers*, Vol. 46, No. 8, pp. 833-854, Aug. 1997.

[Parh00] Parhami, B., *Computer Arithmetic: Algorithms and Hardware Designs*, Oxford, 2000.

[Parh02] Parhami, B., "Number Rerpesentation and Computer Arithmetic," in *Encyclopedia of Information Systems*, Academic Press, to appear in 2002.

[Robe58] Robertson, J.E., "A New Class of Digital Division Methods," *IRE Trans. Electronic Computers*, Vol. 7, No. 3, pp. 218-222, Sep. 1958.

[Zura87] Zurawski, J.H.P. and J.B. Gosling, "Design of a High-Speed Square Root, Multiply, and Divide Unit," *IEEE Trans. Computers*, Vol. 36, No. 1, pp. 13-23, 1987.